Some Gradient Based Joint Diagonalization Methods for ICA

Bijan Afsari, P.S.Krishnaprasad
Institute for Systems Research, University of Maryland
College Park, Maryland 20742, USA
Email:{bijan, krishna}@isr.umd.edu

Abstract

We present a set of gradient based orthogonal and non-orthogonal matrix joint diagonalization algorithms. Our approach is to use the geometry of matrix Lie groups to develop continuous-time flows for joint diagonalization and derive their discretized versions. We employ the developed methods to construct a class of Independent Component Analysis (ICA) algorithms based on non-orthogonal joint diagonalization. These algorithms pre-whiten or sphere the data but do not restrict the subsequent search for the (reduced) un-mixing matrix to orthogonal matrices, hence they make effective use of both second and higher order statistics.

Introduction

- Many problems in Blind Signal Processing can be reduced to approximate Joint Diagonalization (JD) of a set of estimated statistics matrices.
- In the standard ICA model with Gaussian noise:

$$\vec{\mathbf{x}} = A_{n\times n}\vec{\mathbf{s}} + \vec{\mathbf{n}} = \vec{\mathbf{z}} + \vec{\mathbf{n}} \tag{1}$$

for the fourth order matrix cumulant slices of the observed data $\vec{\mathbf{x}}$ we have:

$$\mathrm{Cum}_{\mathbf{x}}(:,:,i,j) = A\Lambda_{ij}A^T, \Lambda_{ij} = \text{diagonal} \tag{2}$$

or equivalently:

$$\forall B \in \mathrm{GL}(n), BA = \text{permuted diagonal} \Rightarrow B\mathrm{Cum}_{\mathbf{x}}(:,:,i,j)B^T = \Lambda \tag{3}$$

where $\mathrm{GL}(n)$ denotes the Lie group of $n \times n$ non-singular matrices and Λ is diagonal matrix.

- (JADE Algorithm) In the absence of noise:
  - Whitening step: Find a whitening matrix W such that $WR_{\mathbf{xx}}W^{-T} = I_{n\times n}$ and whiten $\vec{\mathbf{x}}$ as:

$$\vec{\mathbf{y}} = W\vec{\mathbf{x}} = A_1\vec{\mathbf{s}} \tag{4}$$

We can assume that the unknown matrix A_1 is in the Lie group of $n \times n$ orthogonal matrices $\mathrm{O}(n)$.

  - JD step: Let $\{C_i\}_{i=1}^N$ be a subset of the fourth order cumulant matrix slices of $\vec{\mathbf{y}}$. Find $\Theta \in \mathrm{O}(n)$ such that:

$$\Theta = \arg \min_{B\in \mathrm{O}(n)} J_1(B)$$

where:

$$J_1(B) = \sum_{i=1}^{N} \|BC_iB^T - \mathrm{diag}(BC_iB^T)\|_F^2 \tag{5}$$

and compute the overall un-mixing matrix as $\hat{A} = \Theta W$.

- $\mathrm{O}(n)$ is a compact manifold so $J_1(B)$ has a minimum on it. Therefore J_1 is a suitable cost function for orthogonal JD.
- In the presence of noise the reduced mixing matrix in (4) can not assumed to be orthogonal anymore. What is a suitable JD cost function in the case that the joint diagonalizer is non-orthogonal?.
- A "good" cost function $J(B)$ in terms of the un-mixing matrix B for non-orthogonal JD should be scale invariant as mutual information is:

$$\Lambda = \text{non-singular diagonal}, B \in GL(n) \Rightarrow J(\Lambda B) = J(B) \tag{6}$$

- J_1 defined in (5) is not a suitable cost function for non-orthogonal JD:

$$J_1(\Lambda B) \neq J_1(B) \ \text{ and } J_1(\Lambda B) \to 0 \ \text{ as } \ \|\Lambda\| \to 0 \tag{7}$$

- How can we still use $J_1(B)$ for non-orthogonal JD?.
  Answer: By restricting the gradient of $J_1(B)$ such that it does not reduce the cost function in certain un-wanted directions.

Gradient Flow for Orthogonal Joint Diagonalization

- At any point Θ equip $\mathrm{O}(n)$ with the (Natural) Riemannian metric:

$$\langle \xi, \eta\rangle_\Theta = \mathrm{tr}((\xi\Theta^T)^T\eta\Theta^T) = \mathrm{tr}(\xi^T\eta), \ \ \forall \xi, \eta \in T_\Theta O(n) \tag{8}$$

- A gradient flow for minimization of $J_1(\Theta)$ on $\mathrm{O}(n)$ is given by:

$$\dot{\Theta} = -\Delta\Theta = \sum_{i=1}^{N} [\mathrm{diag}(\Theta C_i\Theta^T), \Theta C_i\Theta^T]\,\Theta, \qquad \Theta(0) = I_{n\times n} \tag{9}$$

where $[X, Y] = XY - YX$ is the Lie bracket.

- Discretization of (9) is not a trivial task, however an Euler scheme with small step-size is promising.

Restricted Gradient Flows for Non-Orthogonal Joint Diagonalization

- Equip the Lie Group $\mathrm{GL}(n)$ with the (Natural) Riemannian metric:

$$\langle \xi, \eta\rangle_B = \mathrm{tr}((\xi B^{-1})^T\eta B^{-1}) = \mathrm{tr}(\eta(B^TB)^{-1}\xi^T), \forall \xi, \eta \in T_BGL(n) \tag{10}$$

- The gradient of $J_1(B)$ with respect to the Riemannian metric defined in (10) is:

$$\nabla J_1 = 4\Delta B, \Delta = \sum_{i=1}^{N} (BC_iB^T - \mathrm{diag}(BC_iB^T))BC_iB^T \tag{11}$$

- Unless $\{C_i\}_{i=1}^N$ have a common joint diagonalizer ∇J_1 can not vanish on $\mathrm{GL}(n)$, i.e. $J_1(B)$ does not have a minimum on $\mathrm{GL}(n)$, which is a result of non-compactness of $\mathrm{GL}(n)$.
- If we restrict the gradient flow for minimization of J_1 such that the cost is not reduced in the directions that correspond to "scaling" we can achieve joint diagonalization.
- **Restriction to SL(n):** Project Δ in (11) to the space of zero trace matrices hence obtain a gradient flow for minimization of J_1 on $\mathrm{SL}(n)$the Lie group of $n \times n$ matrices with unity determinant:

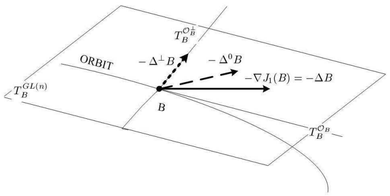
$$\dot{B}(t) = -\Delta^0B(t), \ \ B(0) = I, \Delta^0 = \Delta - \frac{\mathrm{tr}(\Delta)}{n}I_{n\times n} \tag{12}$$

- **A Non-Holonomic Flow**: Project the gradient to the orthogonal complement of the tangent space of the orbit of the left-action of the group of non-singular diagonal on $\mathrm{GL}(n)$, i.e. set the diagonal of Δ to zero:

$$\dot{B}(t) = -\Delta^{\perp}B(t), \ B(0) = I, \ \Delta^{\perp} = \Delta - \mathrm{diag}(\Delta) \tag{13}$$

- Both (12) and (13) are flows on $\mathrm{SL}(n)$ and they do not converge to the global infimum of $J_1(B)$ at $B = 0$:

$$\det(B(t)) = 1 \Longrightarrow \|B(t)\|_2 \geq 1 \tag{14}$$



Discrete Schemes

- (12) and (13) can be discretized by the Euler scheme which is equivalent to steepest descent for small step-size as:

$$B_{k+1} = (I - \mu_kX_k)B_k, \quad B_0 = I \quad k \geq 0 \tag{15}$$

where X_k is computed accordingly.

```
Algorithm 1:
1.set $\mu$ and $\epsilon$.
2.set $B_0 = I_{n\times n}$ or "to a good initial guess".
3.while $\|X_k\|_F > \epsilon$ do
    $B_{k+1} = (I - \mu X_k)B_k$
    if $\|B_{k+1}\|_F$ is "big" then "reduce" $\mu$ and goto 2.
4.end
```

- We need to have large step-size for faster convergence, whereas small step-size to keep $\det(B_k)$ close to one.
- How can we keep the updates on $\mathrm{SL}(n)$ independent of step-size?

An LU Based Discrete Scheme

- Let $\mathcal{L}(\mathcal{U})$ denote the group of lower(upper) triangular matrices with all diagonal elements equal to unity.
- Restrictions of flow (13) to $\mathcal{L}$ and $\mathcal{U}$ are:

$$\dot{U} = -\Delta^{\perp U}U, \ U(0) = I \tag{16}$$

$$\dot{L} = -\Delta^{\perp L}L, \ L(0) = I \tag{17}$$

where $\Delta^{\perp U}$ and $\Delta^{\perp L}$ are the upper and lower triangular parts of $\Delta^{\perp}$, respectively.

- **Observation:** If we discretize the flows in (16) and (17) as in (15) then $\det(U_k) = 1$ and $\det(L_k) = 1$for all k, by construction. This is similar to what happens in Jacobi Rotations.
- Based on the LU factorization of matrices we can consider the joint diagonalization problem as an iterative optimization scheme in which each iteration has two phases: upper and lower triangular joint diagonalization. After each phase the matrices C_i should be updated.

```
Algorithm 2:
Consider the set $\{C_i\}_{i=1}^N$ of symmetric matrices and set $B = I_{n\times n}$.
1. Minimize $J_1(U)$ over $\mathcal{U}$ by using Algorithm 1 to find a solution $U$ to $\dot{U} = -\Delta^{\perp U}U, \ U(0) = I$.
2. set $C_i \leftarrow UC_iU^T$.
3. Minimize $J_1(L)$ over $\mathcal{L}$ by using Algorithm 1 to find a solution $L$ to $\dot{L} = -\Delta^{\perp L}L, \ L(0) = I$.
4. set $C_i \leftarrow LC_iL^T$.
5. set $B \leftarrow L\,UB$
6. if $\|LU - I\|_F$ is "small" end, else goto 1
```

- $\det(B) = 1$ independent of the step-size used.
- Because the updates never leave $\mathrm{SL}(n)$ we can incorporate usual step-selection methods.

A Class of ICA Algorithms Based on Non-Orthogonal JD

- In the presence of noise in (1) after whitening we have:

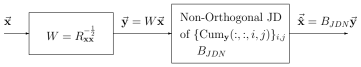
$$\vec{\mathbf{y}} = W\vec{\mathbf{x}} = WA\vec{\mathbf{s}} + W\vec{\mathbf{n}} = A_1\vec{\mathbf{s}} + \vec{\mathbf{n}}_1 \tag{18}$$

where A_1 is only close to orthogonal and its distance to orthogonality depends on the power of noise and the condition number of the mixing matrix.

- **Observations:**
  - The gradient based algorithms developed perform better if the sought matrix is close to orthogonal.
  - Usually by whitening the data the mutual information is reduced so the whitened data is closer to independence.
  - In most cases whitening the data reduces the dynamic range of $\|C_i\|$'s and enables better convergence for numerical methods thereafter.
  - Although estimation of the correlation matrix of $\vec{\mathbf{z}}$ in (1) from observation data $\vec{\mathbf{x}}$ is biased it has less variance than the estimated higher order cumulant slices (this is pronounced especially in small sample sizes). Therefore it is meaningful to use as much information as possible from this correlation matrix provided we can avoid the harm of the "bias" it introduces.
- Based on the above observations we have this class of Non-Orthogonal JD based ICA:

1. Whiten $\vec{\mathbf{x}}$, let W be a whitening matrix, compute $\vec{\mathbf{y}} = W\vec{\mathbf{x}}$ and set $B = W$.
2. Estimate $C = \{C_i\}_{i=1}^N$ a subset of the fourth order cumulant matrix slices of $\vec{\mathbf{y}}$.
3. Jointly diagonalize $C = \{C_i\}_{i=1}^N$ by an orthogonal matrix Θ and set $C_i \leftarrow \Theta C_i\Theta^T$.
4. Jointly diagonalize $C = \{C_i\}_{i=1}^N$ by a non-orthogonal matrix B_{JDN} (using any algorithm such as Algorithms 1 or 2), set $C_i \leftarrow B_{JDN}C_iB_{JDN}^T$ and set $B \leftarrow B_{JDN}\Theta B$.
5. If necessary goto step (3)
6. Compute the recovered signal $\vec{\hat{\mathbf{x}}} = B\vec{\mathbf{x}}$

- Steps (1-3) comprise the JADE algorithm. Orthogonal joint diagonalization can be dropped in most cases.
- Steps 1,2,4 can be summarized as:


$\vec{\mathbf{x}}$
$W = R_{\mathbf{xx}}^{-\frac{1}{2}}$
$\vec{\mathbf{y}} = W\vec{\mathbf{x}}$
Non-Orthogonal JD
of $\{\mathrm{Cum}_{\mathbf{y}}(:,:,i,j)\}_{i,j}$
B_{JDN}
$\vec{\hat{\mathbf{x}}} = B_{JDN}\vec{\mathbf{y}}$

Numerical Simulations

- We compare the performance of the proposed Non-orthogonal JD based ICA algorithms in presence of Gaussian noise with the JADE algorithm.

$$\vec{\mathbf{x}} = A\vec{\mathbf{s}}_{n\times 1} + \sigma\vec{\mathbf{n}} \tag{19}$$

with the assumption that the covariance of noise is identity, σ^2 measures the power of noise.(all random variables are zero mean)

- We estimate an un-mixing matrix B and compute the mixing-unmixing matrix $P = BA$ and use its distance to diagonality up to-permutation as:

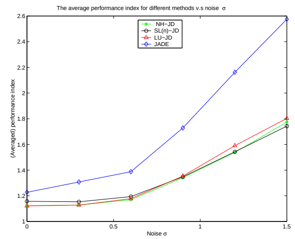
$$\mathrm{Index}(P) = \sum_{i=1}^{n}(\sum_{j=1}^{n}\frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_{j=1}^{n}(\sum_{i=1}^{n}\frac{|p_{ij}|}{\max_k |p_{kj}|} - 1) \tag{20}$$

- The mixing matrix is:

$$A = \begin{bmatrix} -4 & 11 & -1 & 1 & 2 \\ -16 & 11 & 7 & 10 & -13 \\ 1 & 0 & -5 & 0 & 7 \\ 2 & 3 & 21 & 0 & 16 \\ -11 & 1 & -1 & -8 & -6 \end{bmatrix}$$

and sources:

  - Two of them uniform in $[-\frac{1}{2}, \frac{1}{2}]$
  - Two of them two-side exponentially distributed with parameter $\lambda = 1$
  - one is one-side exponential with parameter $\lambda = 1$
- $T = 3500$ data samples are generated for each experiment and for each value of σ
- Three gradient based non-orthogonal JD methods are used for joint diagonalization of the cumulant slices of the whitened data:
  - $\mathrm{SL}(n)$-JD which is an implementation of (12) through Algorithm1 (with $\mu = .01$ and $\epsilon = .01$)
  - NH-JD which is an implementation of (13) through Algorithm1 (with $\mu = .01$ and $\epsilon = .01$)
  - LU-JD which is an implementation of Algorithm 2.(with $\mu = .05$, $\epsilon = .01$ and the LU iteration repeated 5 times)
- As the graph below shows, the non-orthogonal JD based ICA methods proposed outperform JADE in separation performance in the presence of noise:



- These gradient algorithms are slower than JADE, however they require only addition and subtraction.

Conclusion

We introduced gradient based flows for orthogonal and non-orthogonal JD of a set symmetric matrices and developed a family of ICA algorithms based upon non-orthogonal JD. The developed ICA algorithms have the property that after whitening the data they do not confine the search space to orthogonal matrices. This way we can take advantage of both second order statistics (which has less variance) and higher order statistics which are blind to Gaussian noise. Numerical simulations show better performance for the proposed algorithms than for the standard JADE algorithm in Gaussian noise.

- For demos and *MATLAB*® codes please visit the URL: **http://www.isr.umd.edu/Labs/ISL/ICA2004/**